Package ‘modelfree’

August 9, 2012

Type Package

Version 1.1-1

Date 08/08/2012

Title Model-free estimation of a psychometric function

Author Ivan Marin-Franch <imarinfr@indiana.edu>, Kamila Zychaluk <kamila.zychaluk@liverpool.ac.uk>, and David H. Foster <d.h.foster@manchester.ac.uk>

Maintainer Kamila Zychaluk <kamila.zychaluk@liverpool.ac.uk>

Depends R (>= 2.8.1), PolynomF (>= 0.93), SparseM (>= 0.79), stats,utils, base

LazyLoad yes

LazyData yes

Description Local linear estimation of psychometric functions. Provides functions for nonparametric estimation of a psychometric function and for estimation of a derived threshold and slope, and their standard deviations and confidence intervals

URL http://www.modelfree.manchester.ac.uk, http://www.modelfree.liv.ac.uk, http://modelfree.r-forge.r-project.org/

License GPL (>= 2)

R topics documented:

`01_Miranda` *Example 1. Frequency of seeing*

Description

A flash of light of variable intensity was presented repeatedly at a fixed location in the visual field of a subject who reported whether the flash was visible.

Usage

```
data( "01_Miranda" )
```

Format

Data frame with three columns for 10 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Miranda, M. A. \& Henson, D. B. “Perimetric sensitivity and response variability in glaucoma with single-stimulus automated perimetry and multiple-stimulus perimetry with verbal feedback”, Acta Ophthalmologica, 86, 202-206, 2008.

`02_Levi` *Example 2. Visual detection of path deviation*

Description

The subject was presented with the image of a dot moving rightwards on a linear path until it reached the midline of the display, when it changed direction either upwards or downwards. The subject had to indicate the direction.

Usage

```
data( "02_Levi" )
```

Format

Data frame with three columns for 7 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Levi, D. M. \& Tripathy, S. P. “Is the ability to identify deviations in multiple trajectories compromised by amblyopia?”, Journal of Vision, 6(12), 1367-1379, 2006.

`03_Carcagno` *Example 3. Discrimination of pitch*

Description

The subject had to identify the interval containing a tone whose fundamental frequency was different from that in the other two intervals.

Usage

```
data( "03_Carcagno" )
```

Format

Data frame with three columns for 8 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Unpublished data from S. Carcagno, Lancaster University, July 2008

`04_Xie` *Example 4. Discrimination of 'porthole' views of natural scenes*

Description

The subject was presented with a display split into two parts, one containing a pair of patches from the same image, the other a pair from different images, and the subject had to judge which pair came from the same image.

Usage

```
data( "04_Xie" )
```

Format

Data frame with three columns for 10 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Xie, Y. \& Griffin, L. D. "A 'portholes' experiment for probing perception of small patches of natural images", Perception, 36, 315, 2007.

`05_Schofield` *Example 5. Induction of a visual motion aftereffect*

Description

The subject was presented with a moving adaptation stimulus, followed by a test stimulus.

Usage

```
data( "05_Schofield" )
```

Format

Data frame with three columns for 7 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Schofield, A. J., Ledgeway, T., \& Hutchinson, C. V. "Asymmetric transfer of the dynamic motion aftereffect between first- and second-order cues and among different second-order cues", Journal of Vision, 7(8), 1-12, 2007.

`06_Nascimento` *Example 6. Discrimination of image approximations*

Description

The subject was shown an image of a natural scene and an approximation of this image based on principal component analysis. The task was to distinguish between the images.

Usage

```
data( "06_Nascimento" )
```

Format

Data frame with three columns for 8 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Nascimento, S.M.C., Foster, D.H., \& Amano, K. “Psychophysical estimates of the number of spectral-reflectance basis functions needed to reproduce natural scenes”, Journal of the Optical Society of America A-Optics Image Science and Vision, 22 (6), 1017-1022, 2005.

`07_Baker` *Example 7. Auditory detection of a gap in noise*

Description

A 300-ms noise burst containing a gap of 2–8 ms duration or no gap was presented to one ear of a subject.

Usage

```
data( "07_Baker" )
```

Format

Data frame with three columns for 8 stimulus levels: `x`: Stimulus level. `r`: Number of successes. `m`: Number of trials.

References

Baker, R. J., Jayewardene, D., Sayle, C., \& Saeed, S. “Failure to find asymmetry in auditory gap detection”, Laterality: Asymmetries of Body, Brain and Cognition, 13, 1-21, 2008.

bandwidth_bootstrap *Bootstrap estimate of bandwidth*

Description

Finds bootstrap estimate of the optimal bandwidth h for binomial data in local polynomial estimation of psychometric function (PF) with guessing and lapsing rates specified in lims.

Usage

```
bandwidth_bootstrap( r, m, x, H, N, h0 = NULL, link = c( "logit" ), guessing = 0, lapsing = 0, K
```

Arguments

r	number of successes in points x
m	number of trials in points x
x	stimulus levels
H	minimum and maximum values of bandwidth to be considered
N	number of bootstrap replications
h0	pilot bandwidth; if not specified, then the scaled plug-in bandwidth is used
link	name of the link function to be used; default is "logit"
guessing	guessing rate; default is 0
lapsing	lapsing rate; default is 0
K	power parameter for Weibull and reverse Weibull link; default is 2
p	order of the polynomial; default is 1
ker	kernel function for weights; default "dnorm"
maxiter	maximum number of iterations in Fisher scoring; default is 50
tol	tolerance level at which to stop Fisher scoring; default is 1e-6
method	loss function to be used in bandwidth: choose from: 'ISEeta', 'ISE', 'deviance'; default is "all"

Value

h	bootstrap bandwidth for the chosen method; if no method was specified, then it a list of three elements with entries corresponding to the estimated bandwidths on p-scale (h$pscale), on eta-scale (h$etascale) and for mean likelihood (h$deviance)

Examples

```
data("01_Miranda")
h<- bandwidth_bootstrap( example01$r, example01$m, example01$x, c( 0.1, 10 ), 10 )
```

bandwidth_cross_validation *Cross-validation estimate of bandwidth*

Description

Finds the cross-validation bandwidth for the local polynomial estimator of the psychometric function (PF) with guessing and lapsing rates specified in lims.

Usage

```
bandwidth_cross_validation( r, m, x, H, link = c( "logit" ), guessing = 0, lapsing = 0, K = 2, p
```

Arguments

`r` number of successes in points `x`

`m` number of trials in points `x`

`x` stimulus levels

`H` minimum and maximum values of bandwidth to be considered

`link` name of the link function to be used; default is "logit"

`guessing` guessing rate; default is 0

`lapsing` lapsing rate; default is 0

`K` power parameter for Weibull and reverse Weibull link; default is 2

`p` order of the polynomial; default is 1

`ker` kernel function for weights; default "dnorm"

`maxiter` maximum number of iterations in Fisher scoring; default is 50

`tol` tolerance level at which to stop Fisher scoring; default is 1e-6

`method` loss function to be used in cross-validation: choose from: 'ISEeta', 'ISE', 'deviance'; default is "all"

Value

`h` cross-validation bandwidth for the chosen method; if no method was specified, then it a list of three elements with entries corresponding to the estimated bandwidths on p-scale (`h$pscale`), on eta-scale (`h$etascale`) and for mean likelihood (`h$deviance`)

Examples

```
data("01_Miranda")
h<- bandwidth_cross_validation( example01$r, example01$m, example01$x, c( 0.1, 10 ) )
```

bandwidth_optimal *Cross-validation estimate of bandwidth for known distributions*

Description

Finds the cross-validation bandwidth for the local polynomial estimator of the psychometric function (PF) with guessing and lapsing rates specified in lims. The difference between this function and `bandwidth_cross_validation` is that here the true psychometric function is known.

Usage

```
bandwidth_optimal( ptrue, r, m, x, H, link = c( "logit" ), guessing = 0, lapsing = 0, K = 2, p =
```

Arguments

`ptrue`	the true function. Vector with the value of PF at each design point
`r`	number of successes in points `x`
`m`	number of trials in points `x`
`x`	design points
`H`	minimum and maximum values of bandwidth to be considered
`link`	name of the link function to be used; default is "logit"
`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0
`K`	power parameter for Weibull and reverse Weibull link; default is 2
`p`	order of the polynomial; default is 1
`ker`	kernel function for weights; default "dnorm"
`maxiter`	maximum number of iterations in Fisher scoring; default is 50
`tol`	tolerance level at which to stop Fisher scoring; default is 1e-6
`method`	loss function to be used in cross-validation: choose from: 'ISEeta', 'ISE', 'deviance'; default is "all"

Value

`h`	cross-validation bandwidth for the chosen method; if no method was specified,then it a list of three elements with entries corresponding to the estimated bandwidths on p-scale (`h$pscale`), on eta-scale (`h$etascale`) and for mean likelihood (`h$deviance`)

bandwidth_plugin *Plug in estimation of Bandwidth*

Description

Calculates an estimate of the AMISE optimal bandwidth for the local polynomial estimator of a psychometric function.

Usage

```
bandwidth_plugin( r, m, x, link = c( "logit" ), guessing = 0, lapsing = 0, K = 2, p = 1, ker = c
```

Arguments

`r`	number of successes in points `x`
`m`	number of trials in points `x`
`x`	stimulus levels
`link`	link function; default is "logit"
`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0
`K`	power parameter for Weibull and reverse Weibull link; default is 2
`p`	order of the polynomial; default is 1
`ker`	kernel function for weights; default "dnorm"

Value

`h`	plug-in bandwidth (on eta-scale)

Examples

```
data("01_Miranda")
h<-bandwidth_plugin( example01$r, example01$m, example01$x )
```

binomfit_lims *Generalized linear model fit with guessing and lapsing rates*

Description

The function fits a binomial generalised liner model with fixed guessing and lapsing rates

Usage

```
binomfit_lims( r, m, x, p = 1, link = c( "logit" ), guessing = 0, lapsing = 0, K = 2 )
```

Arguments

r	number of successes in points x
m	number of trials in points x
x	stimulus levels
p	degree of the polynomial to be fitted on the linear scale; default is 1
link	link function; default is "logit"
guessing	guessing rate; default is 0
lapsing	lapsing rate; default is 0
K	power parameter for Weibull and reverse Weibull link; default is 2

Value

value	Object with 2 components: b: vector of estimated coefficients for the linear part fit: glm object to be used in evaluation of fitted values

Examples

```
data( "01_Miranda" )
value <- binomfit_lims( example01$r, example01$m, example01$x )
```

binom_g *Psychometric function with guessing rate*

Description

THIS IS AN INTERNAL FUNCTION: USE BINOM_LIMS FOR BEST RESULTS. Maximum likelihood estimates of the parameters of psychometric function with guessing rate (GLM). The estimated parameters for the linear part are in vector b and the estimated guessing rate is guess (GLM).

Usage

```
binom_g( r, m, x, link, p, K, initval )
```

Arguments

r	number of successes in points x
m	number of trials in points x
x	stimulus levels
link	link function
p	degree of the polynomial to be fitted on the linear scale
K	Power parameter in Weibull and reverse Weibull models
initval	initial value for guessing rate

Value

value	Object with 3 components: guessing: estimated guessing rate b: vector of estimated coefficients for the linear part fit: glm object to be used in evaluation of fitted values

Examples

```
data( "01_Miranda" )
value <-binom_g( example01$r, example01$m, example01$x, "logit", 1, 2, 0.01 )
```

binom_gl *Psychometric function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE `BINOM_LIMS` FOR BEST RESULTS. Maximum likelihood estimates of the parameters of psychometric function with guessing and lapsing rates (GLM) or only guessing rate. The estimated parameters for the linear part are in vector b and the estimated limits are in lims.

Usage

```
binom_gl( r, m, x, link, p, K, initval )
```

Arguments

`r`	number of successes in points `x`
`m`	number of trials in points `x`
`x`	stimulus levels
`link`	link function
`p`	degree of the polynomial to be fitted on the linear scale
`K`	Power parameter in Weibull and reverse Weibull models
`initval`	initial value for guessing and lapsing rates

Value

`value`	Object with 4 components: guessing: estimated guessing rate lapsing: estimated lapsing rate b: vector of estimated coefficients for the linear part fit: glm object to be used in evaluation of fitted values

Examples

```
data( "01_Miranda" );
value <-binom_gl( example01$r, example01$m, example01$x, "logit", 1, 2, c( 0.01, 0.01 ) );
```

binom_l *Psychometric function with lapsing rate*

Description

THIS IS AN INTERNAL FUNCTION: USE BINOM_LIMS FOR BEST RESULTS. Maximum likelihood estimates of the parameters of psychometric function with lapsing rate (GLM). The estimated parameters for the linear part are in vector b and the estimated guessing rate is guess (GLM).

Usage

```
binom_l( r, m, x, link, p, K, initval )
```

Arguments

r	number of successes in points x
m	number of trials in points x
x	stimulus levels
link	link function
p	degree of the polynomial to be fitted on the linear scale
K	Power parameter in Weibull and reverse Weibull models
initval	initial value for lapsing rate

Value

value	Object with 4 components: lapsing: estimated lapsing rate b: vector of estimated coefficients for the linear part fit: glm object to be used in evaluation of fitted values

Examples

```
data( "01_Miranda" )
value <-binom_l( example01$r, example01$m, example01$x, "logit", 1, 2, 0.01 )
```

binom_lims *Psychometric function with guessing and lapsing rates*

Description

Maximum likelihood estimates of the parameters of psychometric function with guessing and lapsing rates (GLM) or only guessing rate. The estimated parameters for the linear part are in vector b and the estimated limits are in lims.

Usage

```
binom_lims( r, m, x, gl = c( "both" ), link = c( "logit" ), p = 1, K = 2, initval = NULL )
```

Arguments

r number of successes in points x

m number of trials in points x

x stimulus levels

gl indicator, calulate only guessing if "guessing", only lapsing if "lapsing" and both guessing and lapsing if "both"; default is "both"

link link function; default is "logit"

p degree of the polynomial to be fitted on the linear scale; default is 1

K Power parameter in Weibull and reverse Weibull models; default is 2

initval initial value for guessing and lapsing; default is c(0.01, 0.01) if guessing and lapsing rates are estimated, and 0.01 if only guessing or only lapsing rate is estimated

Value

value The object returned by internal functions binom_g, binom_l, or binom_gl, depending on only guessing, only lapsing, or guessing and lapsing are estimated

Examples

```
data( "01_Miranda" )
value <-binom_lims( example01$r, example01$m, example01$x )
```

binom_revweib *Psychometric function fitting for reverse Weibull link function*

Description

Maximum likelihood estimates of the parameters of the reverser Weibull model(GLM). The estimated parameters for the linear part are in vector b and the estimated exponent is K.

Usage

```
binom_revweib( r, m, x, p = 1, initK = 2, guessing = 0, lapsing = 0 )
```

Arguments

r number of successes in points x

m number of trials in points x

x stimulus levels

p degree of the polynomial to be fitted on the linear scale; default is 1

initK Power parameter in reverse Weibull model; default is 2

guessing guessing rate; default is 0

lapsing lapsing rate; default is 0

Value

`value`	Object with 3 components: b: vector of estimated coefficients for the linear part K: estiamte of the power parameter in the reverse Weibull model fit: glm object to be used in evaluation of fitted values

Examples

```
data( "01_Miranda" )
value <- binom_revweib( example01$r, example01$m, example01$x )
```

`binom_weib` *Psychometric function fitting for Weibull link function*

Description

Maximum likelihood estimates of the parameters of the Weibull model (GLM). The estimated parameters for the linear part are in vector `b` and the estimated exponent is `K`.

Usage

```
binom_weib( r, m, x, p = 1, initK = 2, guessing = 0, lapsing = 0 )
```

Arguments

`r`	number of successes in points x
`m`	number of trials in points x
`x`	stimulus levels
`p`	degree of the polynomial to be fitted on the linear scale; default is 1
`initK`	Power parameter in reverse Weibull model; default is 2
`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0

Value

`value`	Object with 3 components: b: vector of estimated coefficients for the linear part K: estiamte of the power parameter in the reverse Weibull model fit: glm object to be used in evaluation of fitted values

Examples

```
data( "01_Miranda" )
value <- binom_revweib( example01$r, example01$m, example01$x )
```

bootstrap_ci_sl *Bootstrap estimate of confidence interval for slope estimation*

Description

Finds bootstrap estimate of a confidence interval at a significant level alpha for the estimated slope for the local polynomial estimation of psychometric function (PF) with guessing and lapsing rates specified in lims. Confidence interval is based on bootstrap percentiles

Usage

```
bootstrap_ci_sl( TH, r, m, x, N, h0, alpha = 0.05, X = (max(x)-min(x))*(0:999)/999+min(x), link
```

Arguments

`TH` required threshold level

`r` number of successes in points x

`m` number of trials in points x

`x` stimulus levels

`N` number of bootstrap replications

`h0` pilot bandwidth; if not specified, then the scaled plug-in bandwidth is used

`alpha` significance level of the confidence interval

`X` set of value for which to calculate the estimates of PF for the thresholdestimation; if not given 1000 equally spaced points from min to max of xdes are used

`link` name of the link function to be used; default is "logit"

`guessing` guessing rate; default is 0

`lapsing` lapsing rate; default is 0

`K` power parameter for Weibull and reverse Weibull link; default is 2

`p` order of the polynomial; default is 1

`ker` kernel function for weights; default "dnorm"

`maxiter` maximum number of iterations in Fisher scoring; default is 50

`tol` tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`value` Object with 2 components: ci: confidence interval based on bootstrap percentiles sl0: slope estimate

Examples

```
data( "01_Miranda" )
bwd <- 0.2959
value <- bootstrap_ci_sl( 0.5, example01$r, example01$m, example01$x, 10, bwd )
```

bootstrap_ci_th *Bootstrap estimate of confidence interval for threshold estimation*

Description

Finds bootstrap estimate of a confidence interval at a significant level alpha for the estimated threshold for the local polynomial estimation of psychometric function (PF) with guessing and lapsing rates specified in lims. Confidence interval is based on bootstrap percentiles

Usage

```
bootstrap_ci_th( TH, r, m, x, N, h0, alpha = 0.05, X = (max(x)-min(x))*(0:999)/999+min(x), link
```

Arguments

`TH` required threshold level

`r` number of successes in points x

`m` number of trials in points x

`x` stimulus levels

`N` number of bootstrap replications

`h0` pilot bandwidth; if not specified, then the scaled plug-in bandwidth is used

`alpha` significance level of the confidence interval

`X` set of value for which to calculate the estimates of PF for the thresholdestimation; if not given 1000 equally spaced points from min to max of xdes are used

`link` name of the link function to be used; default is "logit"

`guessing` guessing rate; default is 0

`lapsing` lapsing rate; default is 0

`K` power parameter for Weibull and reverse Weibull link; default is 2

`p` order of the polynomial; default is 1

`ker` kernel function for weights; default "dnorm"

`maxiter` maximum number of iterations in Fisher scoring; default is 50

`tol` tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`value` Object with 2 components: ci: confidence interval based on bootstrap percentiles th0: threshold estimate

Examples

```
data( "01_Miranda" )
bwd <- 0.2959;
value <- bootstrap_ci_th( 0.5, example01$r, example01$m, example01$x, 10, bwd );
```

bootstrap_sd_sl *Bootstrap estimate the standard deviation of slope estimation*

Description

Finds bootstrap estimate of the standard deviation of the estimated slope for the local polynomial estimation of psychometric function (PF) with guessing and lapsing rates as specified

Usage

```
bootstrap_sd_sl( TH, r, m, x, N, h0, X = (max(x)-min(x))*(0:999)/999+min(x), link = c( "logit" )
```

Arguments

TH	required threshold level
r	number of successes in points x
m	number of trials in points x
x	stimulus levels
N	number of bootstrap replications
h0	pilot bandwidth; if not specified, then the scaled plug-in bandwidth is used
X	set of value for which to calculate the estimates of PF for the thresholdestimation; if not given 1000 equally spaced points from min to max of xdes are used
link	name of the link function to be used; default is "logit"
guessing	guessing rate; default is 0
lapsing	lapsing rate; default is 0
K	power parameter for Weibull and reverse Weibull link; default is 2
p	order of the polynomial; default is 1
ker	kernel function for weights; default "dnorm"
maxiter	maximum number of iterations in Fisher scoring; default is 50
tol	tolerance level at which to stop Fisher scoring; default is 1e-6

Value

value	Object with 2 components: sd: bootstrap estimate of the standard deviation of the slope estimate sl0: slope estimate

Examples

```
data( "01_Miranda" )
bwd <- 0.2959
value <- bootstrap_sd_sl( 0.5, example01$r, example01$m, example01$x, 10, bwd )
```

bootstrap_sd_th *Bootstrap estimate the standard deviation of threshold estimation*

Description

Finds bootstrap estimate of the standard deviation of the estimated thresholdfor the local polynomial estimation of psychometric function (PF) with guessing and lapsing rates as specified

Usage

```
bootstrap_sd_th( TH, r, m, x, N, h0, X = (max(x)-min(x))*(0:999)/999+min(x), link = c( "logit" )
```

Arguments

TH required threshold level

r number of successes in points x

m number of trials in points x

x stimulus levels

N number of bootstrap replications

h0 pilot bandwidth; if not specified, then the scaled plug-in bandwidth is used

X set of value for which to calculate the estimates of PF for the thresholdestimation; if not given 1000 equally spaced points from min to max of xdes are used

link name of the link function to be used; default is "logit"

guessing guessing rate; default is 0

lapsing lapsing rate; default is 0

K power parameter for Weibull and reverse Weibull link; default is 2

p order of the polynomial; default is 1

ker kernel function for weights; default "dnorm"

maxiter maximum number of iterations in Fisher scoring; default is 50

tol tolerance level at which to stop Fisher scoring; default is 1e-6

Value

value Object with 2 components: sd: bootstrap estimate of the standard deviation of the threshold estimate th0: threshold estimate

Examples

```
data( "01_Miranda" )
bwd <- 0.2959
value <- bootstrap_sd_th( 0.5, example01$r, example01$m, example01$x, 10, bwd )
```

checkinput *Check input parameters of modelfree functions*

Description

THIS IS AN INTERNAL FUNCTION ONLY. Pool of routines that check robustness of input of parameters passed to the other functions of modelfree package.

Usage

```
checkinput(type,x)
```

Arguments

`type`	Type of checking
`x`	Input data to be checked

Value

None. Only stops execution if data is not consistent.

comploglog_link *Complementary log-log link function with guessing and lapsing rates*

Description

Complementaty loglog link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
comploglog_link( guessing = 0, lapsing = 0 )
```

Arguments

`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0

Value

`link`	Complementary log-log link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-comploglog_link( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

comploglog_link_private *Complementary log-log link function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE `COMPLOGLOG_LINK` FOR BEST RESULTS. Complementaty loglog link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
comploglog_link_private( guessing, lapsing )
```

Arguments

`guessing`	guessing rate
`lapsing`	lapsing rate

Value

`link`	Complementary log-log link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-comploglog_link_private( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

deviance2 *Deviance between data and fitted function*

Description

Calculates deviance for data Y and fitted values of the psychometric function f

Usage

```
deviance2(r, m, pfit)
```

Arguments

r	number of successes in points x
m	number of trials in points x
pfit	fitted values

Value

D	Deviance

Examples

```
data( "01_Miranda" )
h = 0.2959
fit <- locglmfit( example01$x, example01$r, example01$m, example01$x, h )
Dev <- deviance2( example01$r, example01$m, fit$fitval )
```

locglmfit *Local generalized linear fitting*

Description

Local polynomial estimator for the psychometric function (PF) and eta function (PF transformed by link) for binomial data; also returns the Hat matrix. Actual calculations are done in `LOCGLMFIT_PRIVATE` or `LOCGLMFIT_SPARSE_PRIVATE` depending on the size of the data set. Here the data are split into several parts to speed up the calculations.

Usage

```
locglmfit( xfit, r, m, x, h, returnH = FALSE, link = c( "logit" ), guessing = 0, lapsing = 0, K
```

Arguments

`xfit`	points in which to calculate the estimate
`r`	number of successes in points `x`
`m`	number of trials in points `x`
`x`	stimulus values
`h`	bandwidths
`returnH`	Boolean; Return or not the hat matrix H? default is `TRUE`
`link`	name of the link function to be used; default is `"logit"`
`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0
`K`	power parameter for Weibull and reverse Weibull link; default is 2
`p`	degree of the polynomial; default `p = 1`
`ker`	kernel function for weights; default `"dnorm"`
`maxiter`	maximum number of iterations in Fisher scoring; default is 50
`tol`	tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`value`	Object with 2 or 3 components: pfit: value of the local polynomial estimate at points xfit etafit: estimate of eta (link of pfit) H: hat matrix (OPTIONAL)

Examples

```
data( "01_Miranda" )
xnew = 1.2 * (0:99)/99+0.1
h <- 0.2959
fit <- locglmfit( xnew, example01$r, example01$m, example01$x, h )
```

`locglmfit_private` *Local generalized linear fitting with usual (non-sparse) matrices*

Description

THIS IS AN INTERNAL FUNCTION: USE `LOCGLMFIT` FOR BEST RESULTS. Fisher scoring method for local polynomial estimator of a psychometric function (PF).

Usage

```
locglmfit_private( xfit, r, m, x, h, returnH, link, guessing, lapsing, K, p, ker, maxiter, tol )
```

Arguments

xfit points in which to calculate the estimate

r number of successes in points x

m number of trials in points x

x stimulus values

h bandwidths

returnH Boolean; Return or not the hat matrix H? default is TRUE

link name of the link function to be used; default is "logit"

guessing guessing rate; default is 0

lapsing lapsing rate; default is 0

K power parameter for Weibull and reverse Weibull link; default is 2

p degree of the polynomial; default p = 1

ker kernel function for weights; default "dnorm"

maxiter maximum number of iterations in Fisher scoring; default is 50

tol tolerance level at which to stop Fisher scoring; default is 1e-6

Value

value Object with 2 or 3 components: pfit: value of the local polynomial estimate at points xfit etafit: estimate of eta (link of pfit) H: hat matrix (OPTIONAL)

Examples

```
data( "01_Miranda" )
xnew = 1.2 * (0:99)/99+0.1
h <- 0.2959
fit <- locglmfit_private( xnew,  example01$r,  example01$m, example01$x, h, FALSE, "logit_link", 0, 0, 2,
```

---

locglmfit_sparse_private

Local generalized linear fitting with sparse matrices

Description

THIS IS AN INTERNAL FUNCTION: USE LOCGLMFIT FOR BEST RESULTS. Fisher scoring method for local polynomial estimator of a psychometric function (PF).

Usage

```
locglmfit_sparse_private( xfit, r, m, x, h, returnH, link, guessing, lapsing, K, p, ker, maxiter
```

Arguments

`xfit`	points in which to calculate the estimate
`r`	number of successes in points `x`
`m`	number of trials in points `x`
`x`	stimulus values
`h`	bandwidths
`returnH`	Boolean; Return or not the hat matrix H? default is `TRUE`
`link`	name of the link function to be used; default is `"logit"`
`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0
`K`	power parameter for Weibull and reverse Weibull link; default is 2
`p`	degree of the polynomial; default `p = 1`
`ker`	kernel function for weights; default `"dnorm"`
`maxiter`	maximum number of iterations in Fisher scoring; default is 50
`tol`	tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`value`	Object with 2 or 3 components: pfit: value of the local polynomial estimate at points xfit etafit: estimate of eta (link of pfit) H: hat matrix (OPTIONAL)

Examples

```
data( "01_Miranda" )
xnew = 1.2 * (0:99)/99+0.1
h <- 0.2959
fit <- locglmfit_sparse_private( xnew,  example01$r,  example01$m, example01$x, h, FALSE, "logit_link", 0,
```

`logit_link` *Logit link function with guessing and lapsing rates*

Description

Logit link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
logit_link( guessing = 0, lapsing = 0 )
```

Arguments

`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0

Value

`link`	Logit link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-logit_link( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

`logit_link_private` *Logit link function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE `LOGIT_LINK` FOR BEST RESULTS.Logit link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
logit_link_private( guessing, lapsing )
```

Arguments

`guessing`	guessing rate
`lapsing`	lapsing rate

Value

`link`	Logit link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-logit_link_private( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

loglog_link — *Log-log link function with guessing and lapsing rates*

Description

Log-log link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
loglog_link( guessing = 0, lapsing = 0 )
```

Arguments

`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0

Value

`link`	Log-log link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-loglog_link( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

loglog_link_private — *Log-log link function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE `CLOGLOG_LINK` FOR BEST RESULTS. Log-log link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
loglog_link_private( guessing, lapsing )
```

Arguments

`guessing`	guessing rate
`lapsing`	lapsing rate

Value

`link`	Log-log link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-loglog_link_private( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

`probit_link` *Probit link function with guessing and lapsing rates*

Description

Probit link for use with GLM functions. The guessing and lapsing rate are fixed to values given in `lims`, hence `link` is a function of only one variable

Usage

```
probit_link( guessing = 0, lapsing = 0 )
```

Arguments

`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0

Value

`link`	Probit link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-probit_link( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

probit_link_private *Probit link function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE PROBIT_LINK FOR BEST RESULTS. Probit link for use with GLM functions. The guessing and lapsing rate are fixed to values given in lims, hence link is a function of only one variable

Usage

```
probit_link_private( guessing, lapsing )
```

Arguments

guessing	guessing rate
lapsing	lapsing rate

Value

link	Probit link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-probit_link_private( 0.1, 0.1 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

revweibull_link *Reverse Weibull link function with guessing and lapsing rates*

Description

Reverse Weibull link for use with GLM functions. The guessing rate and lapsing rate are fixed, and power parameter is set to be equal K, hence link is a function of only one variable

Usage

```
revweibull_link( K, guessing = 0, lapsing = 0 )
```

Arguments

K	power parameter for reverse Weibull link function
guessing	guessing rate; default is 0
lapsing	lapsing rate; default is 0

Value

`link`	Reverse Weibull link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-revweibull_link( 20 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

`revweibull_link_private` *Reverse Weibull link function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE `REVWEIBULL_LINK` FOR BEST RESULTS. Reverse Weibull link for use with GLM functions. The guessing rate and lapsing rate are fixed, and power parameter is set to be equal K, hence link is a function of only one variable

Usage

```
revweibull_link_private( K, guessing, lapsing )
```

Arguments

`K`	power parameter for reverse Weibull link function
`guessing`	guessing rate
`lapsing`	lapsing rate

Value

`link`	Reverse Weibull link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-revweibull_link_private( 20, 0, 0 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

`threshold_slope` *Threshold and slope of estimated psychometric function*

Description

Finds the approximate value of `x (=x_th)` for which the value of the psychometric function is equal thresh and the approximate value of slope in `x_th`

Usage

```
threshold_slope( pfit, xfit, thresh = 0.5 )
```

Arguments

`pfit`	estimated values of the psychometric function
`xfit`	stimulus levels in which the function was estimated
`thresh`	value for which to estimate threshold; default is 0.5

Value

`value`	Object with 2 elements: x_th: estimated threshold slope: estimated value of slope, i.e. derivative of pfit at x_th

Examples

```
data( "01_Miranda" )
xnew = 1.2 * (0:999)/999+0.1
h = 0.2959
fit <- locglmfit( xnew, example01$r, example01$m, example01$x, h )
value <- threshold_slope( fit$pfit , xnew )
```

`weibull_link` *Weibull link function with guessing and lapsing rates*

Description

Weibull link for use with GLM functions. The guessing rate and lapsing rate are fixed, and power parameter is set to be equal K, hence link is a function of only one variable

Usage

```
weibull_link( K, guessing = 0, lapsing = 0 )
```

Arguments

`K`	power parameter for reverse Weibull link function
`guessing`	guessing rate; default is 0
`lapsing`	lapsing rate; default is 0

Value

link	Weibull link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-weibull_link( 20 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

---

weibull_link_private *Weibull link function with guessing and lapsing rates*

Description

THIS IS AN INTERNAL FUNCTION: USE WEIBULL_LINK FOR BEST RESULTS. Weibull link for use with GLM functions. The guessing rate and lapsing rate are fixed, and power parameter is set to be equal K, hence link is a function of only one variable

Usage

```
weibull_link_private( K, guessing, lapsing )
```

Arguments

K	power parameter for reverse Weibull link function
guessing	guessing rate
lapsing	lapsing rate

Value

link	Weibull link for use in all GLM functions

Examples

```
data( "01_Miranda" )
x <- example01$x
r <- example01$r
m <- example01$m
glmdata <- data.frame( cbind( r/m ,m , x ) )
names( glmdata ) <- c( "resp", "m", "x" )
glmformula <- c( "resp ~ x" )
userlink<-weibull_link_private( 20, 0, 0 )
fit <- glm( glmformula, data = glmdata, weights = m, family = binomial( userlink ) )
```

Index